Audio Content Analysis

In the previous chapters, we explained the fundamental methods needed for multimedia content analysis. This chapter explores some of the applications of multimedia content analysis, that pertain to the analysis of speech and music content. In this chapter we will provide a short overview of how typical speech and music analysis systems work by describing on a high level which signal processing and machine learning techniques typically are used. All of the systems presented here will work as presented. However, to achieve high accuracies they require a significant amount of engineering. To go beyond a certain accuracy, research is needed which might redefine how typical systems work in the future (thus potentially making our descriptions obsolete).

Features for Audio Analysis

While machine learning algorithms (as described in Chapter XXX) can usually be used unchanged for different types of input data. The features are that are used as input for machine learning are different because the sensor output for audio is different from vision for example (microphone vs. camera). This section describes commonly used features.

- Energy/Intensity

The most frequently used feature and at the same time most basic audio features is energy often also called intensity. As already described in Chapter XXX. The most common form of energy features for content analysis is obtained by taking the root-mean-square x_{rms} if the n samples x_i :

$$x_{\mathrm{rms}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} {x_i}^2} = \sqrt{\frac{{x_1}^2 + {x_2}^2 + \cdots + {x_n}^2}{n}}.$$

- Pitch

We already described the calculation of pitch in Chapter XXX (speech compression). While pitch alone is rarely used directly for content analysis (except if the goal is to extract pitch), pitch is used as a basis for many features and therefore very important. In speech, detecting voiced and unvoiced regions has high value because unpitched regions are more in many domains more likely to contain noise. The ratio between pitched and unpitched regions (in time) is called Harmonicity-to-Noise Ratio or HNR. HNR can be used as one feature to classify music instruments. It can also be used for detecting a speaker's age as HNR is age dependent as older speakers have a "rougher" voice, i.e. less harmonicity. Often HNR is approximated using a threshold on the zero-crossing rate of the signal (i.e. the number of times the signal changes sign).

- Long-term Average Spectrum (LTAS)

The Long-term Average Spectrum is often used as a feature in speaker identification. It can also be used to classify different recording environments. The LTAS is not a single value but a feature vector. In order to obtain LTAS one calculates the FFT of a signal (see Chapter XXX) and averages the energies in each band over a reasonable amount of time (typical a couple of seconds).

- Formants

Formants are the distinguishing or meaningful frequency components of human speech and also of human singing. The information that humans require to distinguish between vowels can be represented purely quantitatively by the frequency content of the vowel sounds. The formant with the lowest frequency is called f_1, the second f_2, the third f_3, and so on. Usually, f_1 and f_2, are enough to disambiguate a vowel. These two formants determine the quality of vowels in terms of the open/close and front/back dimensions. Thus the first formant f_1 has a higher frequency for an open vowel (such as [a]) and a lower frequency for a close vowel (such as [i] or [u]); and the second formant f_2 has a higher frequency for a front vowel (such as [i]) and a lower frequency for a back vowel (such as [u]). Vowels will almost always have four or more distinguishable formants; sometimes there are more than six. Formants are often measured manually as an amplitude peak in the frequency spectrum of the sound, using a spectrogram. In music processing, formants refer to a peak in the sound envelope and/or to a resonance in sound sources, notably musical instruments as well as that of sound chambers.

Different algorithms are available to track formants. A common way of doing it is to resample the audio signal to a sampling frequency to twice the value of the maximum expected formant (which varies by sex and age, for a young child it could be up to 5500 Hz). Then, LPC coefficients are calculated (see Chapter XXX) and searched for local maxima close to the expected frequency ranges of the formants.

Figure 1 shows a visualization of some of the features discussed thus far, including formants.

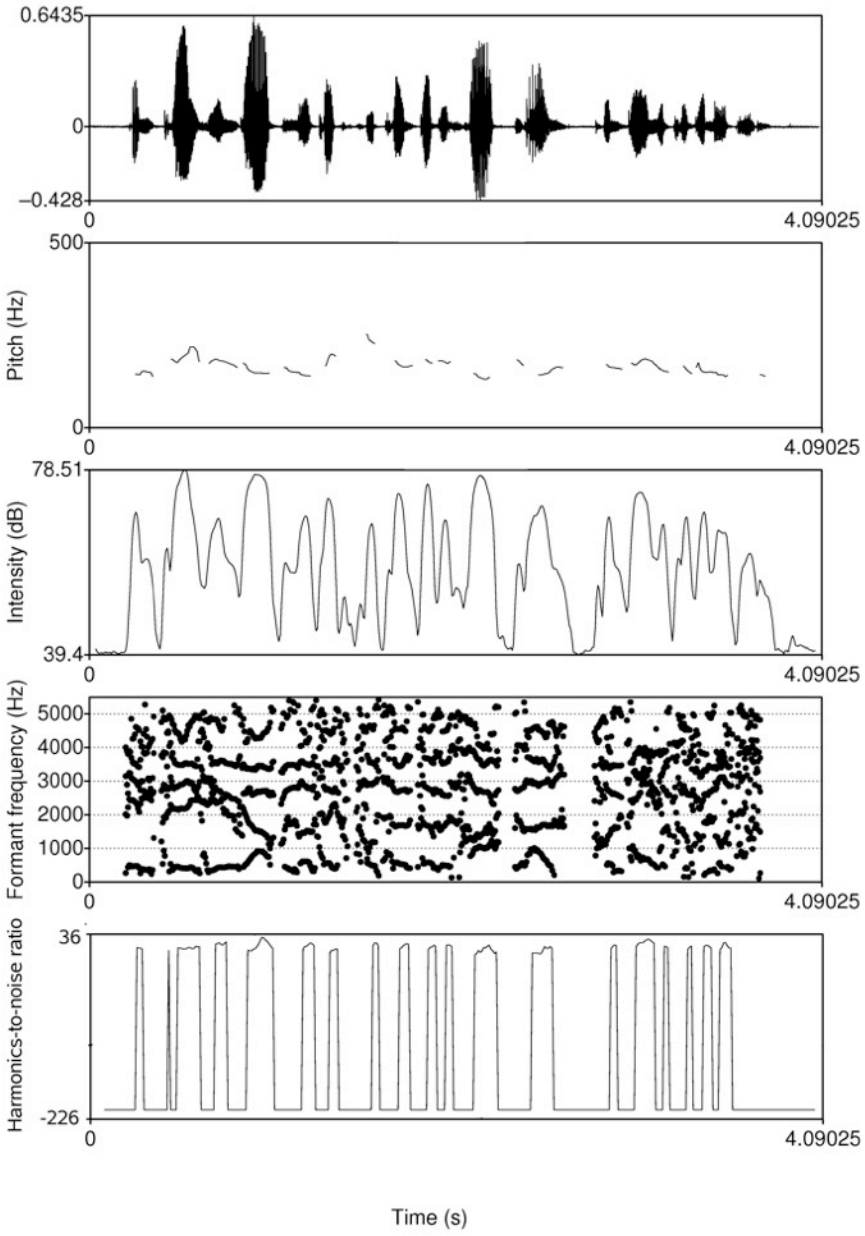

Figure 1. Visualization of some of the features described in this section. (From IEEE TASLP article, co-authored by me)

- Linear Prediction Coefficients (LPC)

The LP coefficients originally developed for speech compression and therefore discussed in Chapter XXX are an often used feature for various speech tasks, such as speech recognition. The coefficients as well as the residual capture the characteristics of different aspects of the signal. LPCs are usually computed on small windows of the signal, for example, 10-30ms. A small window like that is usually called a frame. A frame is the smallest unsplittable unit of analysis.

More frequently used than LP coefficients, however, are the MFC coefficients which are so important that we describe them in their own section.

Mel Frequency Cepstral Coefficients (MFCCs)

Human perception is different between different media. However, it is not clear if the computer should simulate this distinction. Clearly, when machine learning accurately models the human brain, then it would make sense, for example to quantize sensor output logarithmically (see Chapter XXX). However, since current machine learning algorithms are mostly statistical methods, it is not clear if this is beneficial. Nevertheless, some features, such as the Mel Frequency Cepstral Coefficients do incorporate human perceptual properties. In the case of it works remarkably well as MFCCs can be seen as the most frequently used features for any speech analysis tasks, such as speech recognition, speaker identification or language identification. Take a look at Figure 2, which shows the steps involved in the generation of MFCC features as a diagram.

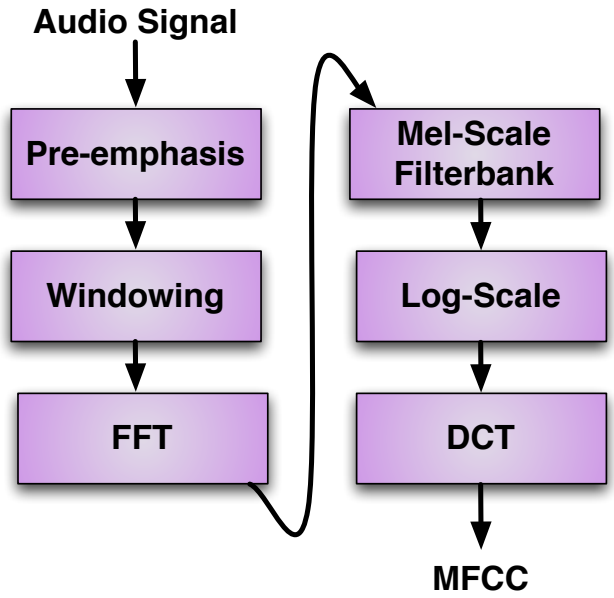

Figure 2. The steps involved in calculating MFCCs.

MFCCs are commonly derived as follows:
- Perform a pre-emphasis of the signal
- Take the Fourier Transform of a windowed excerpt of a signal, typically 10-30ms.
- Map the powers of the spectrum obtained above onto the Mel scale (see Chapter XXX) using triangular overlapping windows.
- Take the logarithm of the powers at each of the Mel frequencies (see Chapter XXX).
- Take the Discrete Cosine Transform (see Chapter XXX0 of the list of Mel log powers, as if it were a signal

The following pseudo-code implements the triangulation-shaped Mel-filter:

```
// Input: samplingrate sr, number of Fourier bins nf, number of Mel-scale
coefficients nm
// Output: A Mel-scale filterbank Matrix M for convolution with an audio
signal.
melfilter(sr, nf, nm)
      nyq := sr/2
      nyq_mel := 2595 * log10(1 + nyq/700.)
      M[nf][nf] := new_zero_matrix()
      FOR i:=0 TO nf-1:
            f := i * nyq / nf
            f_mel := 2595 * log_10(1 + f/700.)
            m_idx := f_mel/nyq_mel*nm
            j=floor(m_idx)
            M(j+1,i) = m_idx-j
            M(j+0,i) = 1.0-m_idx+j
melfilter := M
```

The last DCT step might seem odd to the reader since it means to transform a frequency-space based signal into frequency space. In fact, that's exactly what's being done and the original creators found this so interesting that they named it "cepstrum" as a word-play that exchanges a couple of characters from the word "spectrum". The reason for doing this is that the values obtained in Mel-frequency spectrum is still quite (depending on the window of the FFT). So to further abstract the signal into lower dimensionality, the DCT is used to decorrelate and then reduce the information further, eg. to 12 dimensions (a typical value for speech recognition).

Often, acoustic analysis is performed using "12 dimensional MFCC with delta and delta-deltas". This means, a 12 dimensional DCT is performed as final step and then the differences between the 12 values (deltas) are computed and the differences between the deltas are also computed (delta-delta). This is to approximate the MFCCs and their first and second derivate. For speaker recognition and diarization (see below), often 19 or 22 dimensional MFCCs are computed as the higher dimensions are said to contain more speaker and channel information.

The implementation of MFCC calculation is left as an exercise.

Speech Activity Detection

A very fundamental task in the analysis of an audio signal is to separate human-uttered language from the remaining signal. This function is needed in almost any task that works with language, including speech compression. Typically, however, the methods used in speech compression that detect speech regions in and audio stream are by far not as accurate. Some of them were discussed in Chapter XXX (speech compression), including energy thresholding, and voiced/ unvoiced detection. The biggest challenge with any of the basic methods is the distinction between speech and noise of similar characteristics. Therefore, an approach that usually works better is to build a classifier and train models based on audio files that contain speech in a similar characteristics as the speech to be detected and other models on noise in similar characteristics as the noise to be distinguished from speech. A simple approach, is to train two sets of Gaussian Mixture Models based on 12-dimensional MFCC features that include energy and delta, and

delta-delta coefficients. The decision is usually made on a frame-by frame basis. A HMM in combination with the Viterbi algorithm (see Chapter XXX) can then be used to make an optimal decision for a larger region of the audio stream or file. By training many hours of data into the models current speech activity detectors (often also referred to as speech/non-speech detectors) obtain accuracies of up to 98% when the characteristics of the training data matches that of the evaluation data, e.g. models trained on broadcast TV and applied to broadcast TV or models trained on telephone speech and used in similar phones. The development of a simple model-based speech activity detector is left as an exercise.

Large Vocabulary Automatic Speech Recognition

Speech recognition engines are usually quite large systems. While small-vocabulary speech recognition is used for command and control, e.g. for telephone centers, and pretty much a standard product in industry, state-of-the-art speech recognition engines for large sets of vocabulary and conversational speech contain the work of many many researchers. As a consequence, complete speech recognizers barely exists in universities. Universities usually only deal with certain aspects of the task. It is a domain of companies and research institutes.For that reason and since large vocabulary automatic speech recognition (LVASR) is the most important field in speech processing research, we provide here a rough overview of the functionality of an automatic speech recognizer.

- Feature Extraction

Speech recognition usually starts with several layers of signal processing (e.g., pre-empahsis, windowing, short-term spectral analysis and filtering, and so on). The predominant features used for speech recognition are MFCCs (see above). Although for special purposes, such as high-noise ASR, other features have been designed such as the so-called PLP and RASTA features which are described in research papers the can be found under references.

- Speech Activity Detection

As with most other speech tasks, the first step is a speech activity detection as described above.

- Feature Normalization

After features have been extracted and non-speech is eliminated, the next goal is to try to make the features invariant to anything but the spoken words. Remember that MFCCs are used for various acoustic content analysis tasks. Ideally, we want to eliminate any statistical dependency on the speaker or the channel (microphone, room reverberation). Therefore many techniques exist to normalize features, some are very basic, like Gaussianization, some of them are pretty advanced like Vocal Tract Length Normalization (VTLN). Gaussianization takes a set of audio features and normalizes them so that the histogram of the values forms roughly a Gaussian. This is similar to image histogram equalization (see Chapter XXX) except the target function is a

Gaussian rather than a flat distribution. VTLN is further described in the references to this Chapter.

- Recognition

Now that the audio is filtered so it hopefully contains only speech and features that are invariant to everything but the actual spoken words one uses a classifier, such as a GMMs to compare the spoken words on different levels (using different window length) to the recorded and annotated words in our acoustic models. Usually, a large number of Gaussians that are used in combination to generate likelihoods for particular speech sounds in context are used. The parameters of this acoustic model are then altered further for testing by incorporating one of several related methods for adaptation, for instance Maximum Likelihood Linear Regression (MLLR) (see references). The models are often then trained in a new pass of discriminant learning using techniques like Minimum Phone Error training. In the end, the idea is to recognize "a" by comparing it to all instances of "a" stored in our acoustic model. It is considered to be an "a" if it is very close to all the other "a" and not so close to any other acoustic element, such as "e" or "o" . Using different window length one can compare on sub-phoneme, phoneme, syllable and word level.

- Decoder

Once small-scale recognition (e.g. phonemes, syllables, etc..) is done, the next goal is to glue the pieces together using a so-called language model. The entire acoustic likelihood estimation subsystem is used in combination with a language model probability estimation, which has been trained in a supervised fashion on a large number of words; additionally, there are usually multiple sources of word prediction information (such as large quantities of written text and smaller amounts of transcribed spoken words). Usually HMMs are used to model phoneme and word sequences. For each acoustic instance the recognizer usually outputs a set of alternatives with probabilities, which are used as observations in the HMM. The language model chooses the most likely combination of phonemes, syllables and words, according to the recognizer output. A very hard problem is to handle words that are not part of the language model and usually results in high error rates as surrounding words are also affected.

- Textual Postprocessing

Once decoding is done processing has to be done that may take into account prosody, speech pauses, and other hints to detect sentence boundaries so that punctuation decisions can be made. Also, named entities should be detected so that capitalization works.

This description of automatic speech recognition only conveys the general idea of this class of systems. It shows, however, how the different content analysis and machine learning techniques work together (see Chapter XXX). Speech recognition systems usually contain many signal processing, classification, temporal modeling, and other content analysis tricks that work together.

Speaker Recognition

Speaker recognition is the general term used for acoustic content analysis tasks where the identity of the speaker is to be found automatically by the system. This task has various real-world applications, including forensic analysis, door opening systems, and multimedia retrieval. Depending on the application, there are various “guises” of speaker recognition. Perhaps the most natural form is that of speaker identification, which is to identify the identity of the speaker from a spoken utterance, given the set of possible speakers of that utterance. However, in practical situations it hardly ever occurs that the set of possible speakers is limited, rather, usually there needs to be some verification that the speaker is actually one of the set. Of course, this has to be done after we detected that the audio segment in question is actually speech. Allowing for the possibility of out-of-set speakers is termed open-set speaker identification, and requires that internal similarity scores have some form of “absolute” meaning so that a score can be thresholded, and a hypothesized speaker can be rejected if the score is too low. This capability of rejecting an unknown speaker is so important, that it has been the main focus in speaker recognition methods and its performance evaluation. For non-discriminative modeling, the open-set speaker recognition problem can be generalized to the speaker detection task, where the task is to decide whether or not a given speech segment is spoken by a target speaker. As this general task is at the basis of many different application scenarios, we will use the speaker detection task (equivalent to one speaker open-set identification) as the prototype task in this description.

- Universal Background Model

In order to cope with the open set problem, a Universal Background Model (UBM) is used that represents the speech of "all" possible speakers. It is essentially a GMM consisting of many Gaussians, typical figures are 512–2048 (traditionally, the number of Gaussians are chosen as powers of 2). A UBM is used as denominator in determining a likelihood ratio, representing the likelihood of the “alternative speaker” in speaker detection, i.e. it is used to normalize the score by determining wether the likelihood score obtained by the GMM is typical of a match or might be equally found in two random similar but different speakers. Of course there is no way to represent all possible speakers, yet thousands of speakers are usually used to train the UBM.

In addition to normalizing the score, UBMs are often used as the starting point for modelling a specific speaker, which can be found by adapting the UBM using limited amounts of speech from a specific speaker. It is often the displacement of the centers of the Gaussians that are used to completely characterize a speaker.

- General Architecture

There is a specific training, or enrollment, phase of a speaker, and a testing phase, we can differentiate between the common parts and the training/testing specific parts of the architecture. The common processing steps for a given speech segment are:

1. Speech Activity Detection

2. Feature extraction, usually MFCCs

3. UBM index generation: This step computes the contribution to the UBM likelihood of every Gaussian component, for every frame of the speech segment. Then the indices of the N top-most contributors are extracted, typically N = 5 Gaussians are used. The idea is that these five are enough to compute the likelihood of the frame accurately.

4. Supervector generation: Using the top-N Gaussians per frame in calculation, the means of the UBM can be adapted to maximize the a-posteriori likelihood of the speech segment (so-called MAP adaptation). The shift in means can be said to represent the speaker of the speech segment. A per-dimension scaling of this displacement using the prior and variance parameters of the UBM and concatenation of the scaled displacement vectors into a so-called supervectors allows a geometric interpretation of this space. A speech utterance is represented as a point in this space, and when points lie close together we consider it more likely that the speech was uttered by the same speaker.

The steps specific to training are:

Model generation: There are two distinct classes of modelling used in speaker recognition: generative and discriminative. For a generative model, the MAP-adapted GMM is the model—the important parameters are the (unscaled) means of the Gaussians. Alternatively, a discriminative model can be formed by using a Support Vector Machine (SVM). Additional to the target speaker, for which the model is to be trained, many non-target (i.e., "background") speakers are used to compare the target speaker to. As described in Chapter XXX, the SVM tries to maximize the margin between the target speaker and the background speakers. That is, it tries to position a hyperplane in supervector space which has a maximum distance from the target speaker. The SVM model now is characterized by the normal n of this separating hyperplane and an offset, 500–2000 background speakers are used typically.

Z-norm statistics collection: For generative modeling, a set of background speakers can be used in a different way. The likelihoods of background speakers given the target-speaker GMM can be calculated for a set of non-target speakers. The mean and variance of these likelihoods can be stored with the speaker model, and used for score normalization in the test phase. This is known as Z-norming.

- Evaluation Metrics

Applications in speaker detection range from target-sparse applications in intelligence (finding the few utterances from a target speaker in a very large database of recordings) to target-rich applications such as access control (finding the presumably very few break-in attempts in long sequences of genuinely authorized speakers). In a detection trial, the prior probability of a target speaker plays a crucial role. However, these priors cannot be determined by the speaker

recognition technology itself, and are given by the application. Therefore, the framework in which a speaker recognition system is evaluated is by a defining a cost function:

$$C_{\text{det}} = C_{\text{miss}} P_{\text{tar}} P_{\text{miss}} + C_{\text{FA}}(1 - P_{\text{tar}}) P_{\text{FA}}$$

Here, the application-specific cost parameters C_{miss} and C_{FA} determine the expected costs made in decision errors. The error rates P_{FA} and P_{miss} indicate the probability of a miss (a not-detected target trial) and false alarm (a falsely detected non-target trials), and must be determined in an evaluation of the system. It can be seen that the target prior P_{tar} governs the cost function.

Acoustic Event Detection

Acoustic event detection (AED) identifies different acoustic events inside and audio stream. The task is inherently harder than speech activity detection because the different event classes can have severely different or similar properties. Often it is hard to model varying durations (even inside the same class of events) and, of course, it is not guaranteed that the sound for a particular event is not a subset of another one (this is a similar problem as in entropy-based compression, see Chapter XXX). AED systems are therefore trained on a case-by-case basis with many hours of data. They are very similar to speaker recognition systems and baseline approaches use Gaussian Mixture Models (GMMs) combined with Hidden Markov Models (HMMs) using a Universal Background Model (UBM). However, recently research has shown that so-called supervector methods can improve event detection. Recent approaches of acoustic event detection therefore compute the mean and standard deviations of the feature trajectories, and use these statistics as input features for a Support Vectore Machine (SVM). This "GMM-SVM" approach combines the discriminative properties of SVMs with the ability of GMMs to deal with variable length sequences. One can use a linear kernel derived from Kullback-Leiber distance for this:

$$K_{lin}(s_a, s_b) = \sum_{i=1}^{M} (\sqrt{w_i}\Sigma_i^{-\frac{1}{2}}\mu_i^a)(\sqrt{w_i}\Sigma_i^{-\frac{1}{2}}\mu_i^b)^t$$

where s_k is a GMM supervector obtained by pooling together all the Gaussian means μ_{ki} of a means-only MAP-adapted GMM for the sequence k. Σ_i and w_i are the original weight and covariance of each Gaussian on the UBM model used for adaptation. A more detailed description of the approach can be found in the articles referenced.

Speaker Diarization

The goal of speaker diarization is to segment a single-channel audio recording into speaker-homogeneous regions, and cluster these, with the goal of answering the question "who spoke when?" [22]. Figure 3 illustrates the idea. Speaker diarization has a large set of possible and actual applications. Usually, it is used as a front-end (also called upstream) application for different higher level tasks, such as speech recognition, meeting, seminar, or broadcast news

navigation, or even dominance detection (based on clues such as who speaks most, who interrupts whom).

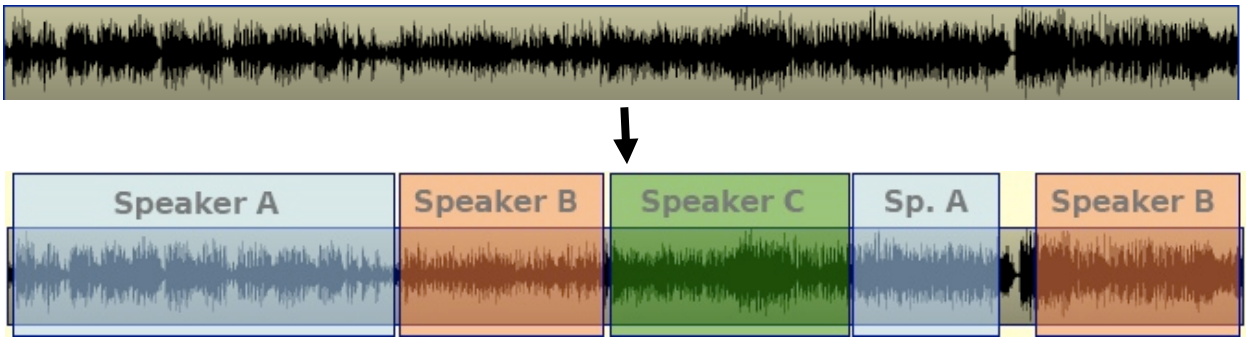

Figure 3. The task of Speaker Diarization is to determine "who spoke when" whithout any prior knowledge about the content of the audio track.

In contrast to speaker recognition or identification, speaker diarization attempts to use no prior knowledge of any kind. This usually means that no specific speaker models are trained for the speakers that are to be identified in the recording. In practice this means a speaker diarization system has to answer the following questions:
- What are the speech regions?
- How many speakers occur in the recording?
- Which speech regions belong to the same speaker?

Therefore, a speaker diarization system conceptually performs three tasks: First, speech activity detection, second, detect speaker changes to segment the audio data, third, group the segmented regions together into speaker-homogeneous clusters. Some systems unify the two last steps into a single one, i.e., segmentation and clustering is performed in one step. Over the years, many different algorithms have been developed in the speech research community.

Many state-of-the-art speaker diarization systems, including the ICSI Speaker Diarization engine (see below), use a one-stage approach, i.e., the combination of agglomerative clustering with Bayesian Information Criterion (BIC) (see Chapter XXX) and Gaussian Mixture Models (GMMs, see Chapter XXX) of frame-based cepstral features (MFCCs, see above).

In two-stage speaker diarization approaches, the first step (speaker segmentation) aims at detecting speaker change points and is essentially a two-way classification/decision problem, i.e., for each frame, a decision needs to be made on whether this is a speaker change point or not. After the speaker change detection, the speech segments, each of which contains only one speaker, are then clustered using either top-down or bottom-up clustering. In model-based approaches, pre-trained speech and silence models are used for segmentation. The decision about speaker change is made based on frame assignment, i.e. the detected silence gaps are considered to be the speaker change points. Metric-based approaches are more often used for speaker

segmentation. Usually, a metric between probabilistic models of two contiguous speech segments, such as Gaussian Mixture Models, is defined and the decision is made via a simple thresholding procedure. To provide some more technical details about, how a diarization system actually works, we describe one actual system as an example. More details can be found in the original research papers presented under references.

The audio track is usually processed as 19th-order MFCC features using a frame size of 10 ms. A speech activity detector (see above) is used to filter out regions that do not contain speech. The non-speech regions are excluded from the segmentation and clustering. The algorithm is initialized using a much higher amount of clusters than speakers expected in the audio track. Let this number be k. An initial segmentation is generated by randomly partitioning the audio track into k segments of the same length. Using the initial segmentation, k Gaussian Mixture Models are trained. As classifications based on 10 ms frames are very noisy, a minimum duration of 2.5 seconds is assumed for each speech segment. A majority vote is then used to combine the individual decisions. The algorithm then performs the following loop:

Re-Segmentation: Compute the likelihoods with respect to each Gaussian Mixture Model and vote to determine the assignment of each minimum duration segment to a particular model.

Re-Training: Given the new segmentation of the audio track, train new Gaussian Mixture Models for each of them.

Cluster Merging: Given the new Gaussian Mixture Models, try to find the two models that most likely represent the same speaker. This is done by computing the BIC score (Bayesian Information Criterion) of each of the models and the BIC score of a new GMM trained on the merged segments for two clusters. If the BIC score of the merged Gaussian Mixture Model is smaller than or equal to the sum of the individual BIC scores, the two models are merged and the algorithm loops at the re-segmentation step using the merged Gaussian Mixture Model. If no pair is found, the algorithm stops.

The output of a speaker diarization system consists of metadata describing speech segments in terms of starting time, ending time, and speaker cluster name. This output is usually evaluated against manually annotated ground truth segments. A dynamic programming procedure is used to find the optimal one-to-one mapping between the hypothesis and the ground truth segments so that the total overlap between the reference speaker and the corresponding mapped hypothesized speaker cluster is maximized. The difference is expressed as Diarization Error Rate which is defined by the US National Institute of Standards and Technology (NIST). The Diarization Error Rate (DER) can be decomposed into three additive components: misses (speaker in reference, but not in hypothesis), false alarms (speaker in hypothesis, but not in reference), and speaker-errors (mapped reference is not the same as hypothesized speaker). The difference is expressed as Diarization Error Rate (DER) which is defined as follows:

$$DER = \frac{\sum_{s=1}^{S} \mathrm{dur}(s) \cdot (\max(N_{ref}(s), N_{hyp}(s)) - N_{correct}(s))}{\sum_{s=1}^{S} \mathrm{dur}(s) \cdot N_{ref}}$$

with S being the total number of speaker segments where both reference and hypothesis files contain the same speaker pair(s). It is obtained by comparing the hypothesis and reference speaker turns. The terms $N_{ref(s)}$ and $N_{sys(s)}$ indicate the number of speakers speaking in segment s, and $N_{correct(s)}$ indicates the number of speakers that speak in segment s and have been correctly matched between reference and hypothesis. Segments labelled as non-speech are considered to contain 0 speakers. DER is usually expressed in %. When all speakers and the non-speech in a file are correctly matched the error is 0%.

Speaker Diarization is currently an area of research. Different approaches are investigated, including methods that incorporate spatial information such as video images or the time delay of arrival from different microphones. Research problems include that many speaker diarization systems are not robust enough to be easily ported across different task and data domains. Often parameters of systems are tuned to a particular set of data such as broadcast news or meetings. In a new domain, tuning of parameters often starts from scratch. Also, even variations inside one domain e. g., meeting data recorded at different sites can lead to large variations in performance. Speaker variations caused by emotions or very short interruptions (e. g., shorter than the minimum duration constraint) pose challenges that are yet to be addressed, possibly by multimodal approaches. The greatest challenge is the handling of overlapped speech, which needs to be attributed to multiple speakers.

Other tasks

This chapter only exemplified a couple of acoustic analysis tasks. Many other problems exist, especially in the musical domain, and solutions to them are evolving rapidly as the demand for working searching, organizing, and editing multimedia content creates new challenges to the research community. Chapter XXX will elaborate on Multimedia Information Retrieval, which uses many of the tools presented in the this chapter and in the chapter of visual analysis (see Chapter XXX).

Literature

- L. R. Rabiner and B. H. Juang, *"Fundamental of Speech Recognition"*, Prentice hall, 1993
- B. Gold and N. Morgan: "*Speech And Audio Signal Processing: Processing And Perception Of Speech And Music*" (Paperback), Wileys and Sons, August 2006.
- Müller, C.(ed.) (2007). "*Speaker Classification I - Fundamentals, Features, and Methods*", Springer, New York - Berlin.

Research Articles

- James Bergstra, Michael Mandel, and Douglas Eck. Scalable genre and tag prediction with spectral covariance. In *Proceedings of the 11th International Conference on Music Information Retrieval (ISMIR)*, pages 507-512, August 2010.
- K. Boakye, B. Trueba-Hornero, O. Vinyals, and G. Friedland, "Overlapped speech detection for improved speaker diarization in multiparty meetings," Proc. ICASSP, pp. 4353–435.
- Chen, S. S. and Gopalakrishnan, P., "Clustering via the bayesian information criterion with applications in speech recognition," Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, Vol. 2, Seattle, USA, pp. 645-648.
- J. Cohen, T. Kamm, and A. Andreou, "Vocal tract normalization in speech recognition: compensation for system systematic speaker variability," J. Acoust. Soc. Am., vol. 97, no. 5, Pt. 2, pp. 3246–3247, 1995.
- G. Friedland, O. Vinyals, Y. Huang, and C. Mueller, "Prosodic and Other Long-term Features for Speaker Diarization," IEEE Transactions on Audio, Speech, and Language Processing, Vol 17, No 5, pp 985–993, July 2009.
- D.A. Van Leeuwen and M. Konec˘ny´, "Progress in the AMIDA Speaker Diarization System for Meeting Data," in Multimodal Technologies for Perception of Humans: International Evaluation Workshops CLEAR 2007 and RT 2007, Baltimore, MD, USA, May 8-11, 2007, Revised Selected Papers. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 475–483.
- J. Ramirez, J. M. Girriz, and J. C. Segura, "Voice activity detection. Fundamentals and speech recognition system robustness," in Robust Speech Recognition and Understanding, M. Grimm and K. Kroschel, Eds., Vienna, Austria, June 2007, p. 460.
- S. Wegmann and L. Gillick, "Why has (reasonably accurate) Automatic Speech Recognition been so hard to achieve?" Tech. Report, Nuance Communications, http://web.mit.edu/kenzie/www/wegmann/wegmann_gillick_why.pdf, 2009.
- S. Tranter and D. Reynolds, "An overview of automatic speaker diarization systems," IEEE TASLP, vol. 14, no. 5, pp. 1557–1565, 2006.

Exercises

1. Create a program that calculates MFCC features and visualizes them. Input different audio events and notice how the features change.
2. Implement a simple speech activity detection as described in the chapter. Use available corpora from the Internet to train your Gaussians. Measure the classification error with a) different set of parameters when training the Gaussians b) the number of Gaussians c) when testing on the training set d) when testing on a different audio corpus.
3. Discuss possibilities to extend a speech recognition system with video analysis. When do you expect your multimodal system to work well?
4. Explain how you would like to change the behavior of a speaker identification system for these applications: Video retrieval, biometric authorization, a game that gives you a score based on how good you imitate a celebrity's voice.
5. Describe an alternate clustering algorithm for Speaker Diarization that starts with one cluster and splits sub-sequently. Analyze the runtime for the algorithm presented in this chapter and your new one.

6. In the segmentation/clustering algorithm presented in this chapter, the clusters are said to be “purified” in each step by merging two clusters according to the BIC. Provide a colloquial explanation for how this “purification” works. Explain possible problems.
7. Perform the following experiment: Ask a co-student/co-worker to find a video on the Internet in a language that you do not speak and where you do not know the participants. It should contain a conversation of several minutes with at least 4 speakers (a foreign talk show might be a good choice). Do not watch the video, only listen to the audio and perform manual speaker online diarization by saying ”speaker 1”, ”speaker 2”. Let your co-worker/co-student rate you: How good are you at assigning the right speakers in a normal and in an overlap situation? How does the situation improve once you look at the video?
8. Pick one of the audio analysis tasks described above. Explain typical expected problems when performing the task as presented here in the following data domains: Recorded voice-over-IP phone conference, a board meeting recorded with a microphone array, a conversation recorded with a cell-phone in a car, a recorded theater performance, broadcast news, an air-traffic control session, a microphone mounted onto a surveillance camera.